

CS Optional Standards Change Log - Winter 2026

Domain	Grade	Standard	Concern Identified	Change Details	Revised Language
Algorithms	K–2	K-2.ALGM.01 Standards Statement Model daily processes by creating and following algorithms that include sequence, events, and iteration to complete tasks.	Sequencing in text-based programs unclear	Specify sequencing proceeds left-to-right and top-to-bottom	Model daily processes by creating and following algorithms that include sequencing (proceeding left-to-right and top-to-bottom), events, and iteration to complete tasks.
Algorithms	3–5	3-5.ALGM.02 Standards Statement Modify algorithms with repeating patterns to use iteration instead of repeated instructions.	Strong focus on requirements not reflected elsewhere	Use as model; introduce and spiral requirements definition and feedback across grade bands	Modify and model algorithms with repeating patterns to use iteration instead of repeated instructions such as spiral requirements .
Algorithms	3–5	3-5.ALGM.04 Standards Statement Assess an algorithm's effectiveness in solving a problem	"Solving a problem" is vague	Replace with "meets specific conditions and does what the user expects"	Assess whether an algorithm meets specific conditions and does what the user expects , using defined requirements and feedback to evaluate and improve the solution.
Algorithms	6–8	6-8.ALGM.04 Standards Statement Demonstrate the correctness of algorithms for given inputs.	Algorithm correctness underspecified	Require students to demonstrate how algorithms meets conditions and user expectations	Demonstrate how an algorithm meets defined conditions and user expectations for given inputs.

<p>Algorithms</p>	<p>High School</p>	<p>HSF.ALGM.02 Standards Statement Describe the differences between deterministic algorithms and probabilistic algorithms.</p>	<p>Missing core CS concepts</p>	<p>Add algorithmic complexity (Big-O), heuristics, parallel, and non-deterministic algorithms</p>	<p>Describe and analyze differences among deterministic algorithms and probabilistic algorithms, accounting for increasing complexity.</p>
<p>Programming</p>	<p>K–12</p>	<p>Multiple PRG standards (ODE Selected) Adjusted 3-5.PRG.01,02, & 04 to include name procedures and grouping steps</p> <p>3-5.PRG.01 Develop code from a student-created algorithm that includes sequence, events, iteration, and selection to express ideas or complete a task.</p> <p>3-5.PRG.02 Debug errors in a program that includes sequence, events, iteration, and selection</p> <p>(ODE Selected) Adjusted 6-8.PRG 01, 02, 05 to include procedures with parameters and generalizing behavior.</p> <p>6-8.PRG.01 Develop code from student-created algorithms that include sequence, events, iteration, selection, and variables to express ideas, complete a task, or solve a problem.</p>	<p>Missing emphasis on core abstractions</p>	<p>Introduce and spiral procedures/functions, parameters, APIs, and abstraction</p>	<p>Remove K-2 PRG standard</p> <p>3-5.PRG.01 (Revised) Develop code from a student-created algorithm that includes sequence, events, iteration, and selection, and organize code into reusable blocks or simple procedures to reduce repetition or complete a task</p> <p>3-5.PRG.02 (Revised) Debug errors in a program that includes sequence, events, iteration, selection, and reusable code blocks or procedures.</p> <p>6-8.PRG.01 (Revised) Develop code from student-created algorithms that include sequence, events, iteration, selection, variables, and procedures with parameters to organize and abstract program functionality.</p> <p>6-8.PRG.02 (Revised) Analyze how a segment of code works by identifying procedures, parameters, variables, and their roles in implementing abstraction.</p> <p>6-8.PRG.05 (Revised)</p>

		<p>6-8.PRG.02 Analyze how a segment of code works by identifying and describing the roles of key components.</p> <p>6-8.PRG.05 Utilize reference documentation, online resources, and programming tools to assist in writing, debugging, and improving code.</p> <p>(ODE Selected) Adjusted HS.PRG.02, 03, 04 to include modular architecture, API's, and designing systems with abstraction layers.</p> <p>HSF.PRG.02 Analyze the purpose of a segment of code.</p> <p>HSF.PRG.03 Assess opportunities to incorporate external code, utilizing documentation, libraries, APIs, development tools, and online resources.</p> <p>HSF.PRG.04 Modify a program to improve or change functionality, usability, accessibility, safety, accuracy, or inclusivity of a program.</p>			<p>Use reference documentation, libraries, APIs, and development tools to implement prewritten functions or services within a program.</p> <p>HSF.PRG.02 (Revised) Analyze a segment of code to explain how procedural abstraction, parameters, and modular design contribute to functionality and readability.</p> <p>HSF.PRG.03 (Revised) Evaluate and integrate external libraries or APIs using documentation, assessing how abstraction enables code reuse and interoperability</p> <p>HSF.PRG.04 (Revised) Refactor or redesign a program to improve functionality, usability, accessibility, safety, accuracy, or inclusivity through improved abstraction and modularization.</p>
Programming	3–5	<p>3-5.PRG.04 Standards Statement Collaborate with a team to create a program, ensuring that all team members have a role and</p>	Teamwork introduced but not extended	Consider adjusting to remove teamwork concept. Is this a CS standard or an employability skill?	Collaborate with peers to design, implement, document, and review a programming project that uses simple abstraction (e.g., named procedures) to express an idea or solve a problem.

		contribute equally.			
Programming	6–8	6-8.PRG.06 Standards Statement Use standard practices to test, debug, document, and peer-review code	Redundant bullets on tracing/debuggers	Combine bullets; clarify debuggers trace code step by step	Use standard practices to test, debug by tracing , document, and peer-review code.
Programming	6–8	6-8.PRG.07 Standards Statement Modify a program to improve usability and accessibility.	Incorrect term and missing usability focus	Replace “accessibility” with “readability”; add usability testing	Modify a program to improve readability and usability testing .
Programming	6–8	6-8.PRG.08 Standards Statement Apply inclusive collaboration practices to support all stages of programming, from planning to testing	“Collaborative debugging” misrepresents practice	Replace with “Asking for help”	Recommend no change
Programming	High School	HSF.PRG.05 Standards Statement Argue how a program does and does not address a given problem.	“Argue” is adversarial	Replace “Argue” with “Discuss”	Discuss how a program does and does not address a given problem.
Data & Analysis	3–5	3-5.DAA.03 Standards Statement Design a data collection approach that addresses the needs of people from different backgrounds or groups	Viewed as incorrect and/or harmful	Remove standard entirely	Design a data collection approach that thoughtfully considers and accommodates the diverse needs, perspectives, and experiences of individuals from a variety of backgrounds and groups .

Data & Analysis	6–8	6-8.DAA.02 Standards Statement Analyze how a segment of code works by identifying and describing the roles of key components.	Claim that “too much data” overwhelms is misleading	Revise/remove bullet; emphasize sufficient and statistically significant data	Analyze how a segment of code works by identifying and describing the roles of sufficient and statistically significant data
Data & Analysis	All	DAA domain (ODE Selected) Adjusted 6-8 DAA.01 to emphasize statistics and focus on data types	Overemphasis on statistics & bias is discussed without causes or evaluation	Broaden to spatial, audio, visual, text, and network data; consider renaming to “Data and Computing” Add producer motives, confirmation bias, correlation vs causation, data validity	Add K-2.DAA.04 Recognize that data may represent some people or perspectives more than others and discuss how missing or incomplete information can affect conclusions. 3-5.DAA.04 Explain design choices, identify limitations in the data, describe who produced the data and why, and distinguish between observed patterns and claims about cause and effect. 3-5.DAA.05 Consider whose perspectives may be missing, how question wording or measurement methods may introduce bias, and how personal assumptions or confirmation bias may influence interpretation. 6-8.DAA.01 (Revised) Collect and generate multiple types of data—quantitative, qualitative, spatial, audio, visual, text, and network—using computational tools.

Computing Systems & Society	3–5	3-5.CSS.02 Standards Statement Explain how computing devices connect to the Internet using wires or wireless signals.	Missing common technologies	Add Bluetooth and GPS examples	Explain how computing devices such as Bluetooth and GPS connect to the Internet using wires or wireless signals.
Computing Systems & Society	6–8	6-8.CSS.01 Standards Statement Describe the structure and organization of file systems, including file naming conventions, directories, and file permissions, and their role in storing and managing data	Outdated analogy; missing structure	Remove filing cabinet analogy; teach tree hierarchies; specify read/write/execute permissions	Recommend no change
Computing Systems & Society	6–8	6-8.CSS.03 Standards Statement Explain the key components of the Internet, their roles and functionality, and how they contribute to its resilience.	Obsolete technology review lacks value	Remove standard	Explain the foundational architecture of the Internet, including its core components, governing protocols, and distribution methods Add- 6-8.CSS.06 Analyze how the elements of governing protocols, and distribution methods enable scalability, compatibility, and resilience surrounding the internet in a changing technological landscape.
Computing Systems & Society	High School	HSF.CSS.05 Standards Statement Investigate the societal and environmental impacts of computing systems and the physical infrastructure that supports them.	Internet infrastructure incomplete	Explicitly include Internet Service Providers (ISPs)	Recommend no change

Computing Systems & Society	High School	HSF.CAS.01 Standards Statement Differentiate major eras in computing history and key advancements by notable individuals and organizations.	No clear instructional value	Remove standard	Differentiate major eras in computing history and key advancements by notable individuals and organizations and how they have led to present day systems.
Computing Systems & Society	High School	HSF.CAS.02 Standards Statement Evaluate policies and legislation designed to encourage ethical innovation and minimize societal risks associated with technology.	Policies lack rationale	Add discussion explaining why policies are needed	Evaluate policies and legislation designed to encourage ethical innovation and minimize societal risks associated with technology and why these policies are necessary.
Cybersecurity / Digital Well-being	All	Multiple CSS/CAS standards	Missing practical personal security instruction	Add secure password creation, password management, personal data risks, screen addiction	Add K-2.CSS.06 Create a simple, strong password using a combination of letters, numbers, and symbols (with teacher guidance). K-2.CSS.07 Explain why passwords should be kept private (except with a trusted adult). 3-5.CSS.07 Recognize examples of personal information (name, address, school, birthday) that should not be shared online without adult permission. 3-5.CSS.08 Describe basic password management strategies (e.g., using a password manager with adult guidance, not sharing passwords with friends).
Cybersecurity / Digital Well-being	K-2	Multiple domains	No AI exposure	Add identification of everyday AI; discuss helpful vs harmful	On Pg. 5- Rewrite final bullet under implementation of standards-Some high schools offer classes in computer science that address more specific topics such as

					<p>Cybersecurity, Artificial Intelligence, robotics, or game design. Many of those courses are part of CTE Programs of Study. The standards for more advanced or specialized courses are not included in this document. Educators seeking guidance on content in advanced courses may wish to refer to the Oregon Statewide Program of Study Framework: Information and Communication Technology Career Cluster, CSTA Reimagining CS Pathways, or the national CSTA computer science standards.</p>
Artificial Intelligence	3–5	Multiple domains	No explicit AI learning	Add simple AI tools, limitations, and bias exploration	See change from box above
Artificial Intelligence	6–8	Multiple domains	AI limited to ethics	Add data-driven vs rule-based systems; simple model training	See change from box above
Artificial Intelligence	High School	Multiple domains	AI mostly conceptual	Add building, testing, and evaluating AI systems; consider standalone AI strand	See change from box above
Feasibility	K–8	Multiple standards	Developmentally inappropriate and unrealistic	Reduce abstraction; narrow scope; integrate with existing subjects; acknowledge constraints	This feedback contradicts other feedback we got. Is this a DOK question in which we need to emphasize that a student can learn these standards at this level?

					Language for Adrienne to draft-Support and resources to be coming to help teachers work.
Alignment	All	Entire framework	Partial CSTA adoption risks incoherence	Consider adopting CSTA 3.0 more comprehensively; include CSTA pillars in preface	No change at this time
Coding	All	Every standard	N/A	Based on feedback from the standardizing standards work group	Adding our content standards code to the beginning of each standard- Ex: K-2.ALGM.01 would become CS.K-2.ALGM.01